Content Based Image Retrieval

By

Shreyal C. Shah
20BCE523

Jainil A. Solanki
20BCE526

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A+' GRADE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

Content Based Image Retrieval

Report

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Science & Engineering

By

Shreyal C. Shah
20BCE523

Jainil Solanki
20BCE526

Guided By
Prof Zunnun Sir
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad 382481

CERTIFICATE

This is to certify that the Internship entitled Associate Software Developer submitted by Shreyal Shah(20BCE523) & Jainil Solanki(20BCE526), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof. Zunnun Sir
Computer Science and Engineering Dept.,
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Madhuri Bhavsar,
Professor and HOD,
Computer Science and Engineering Dept.,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGEMENT

First off, We want to express my gratitude to Prof. Zunnun Sir, professor at Nirma University, for providing us with the chance to do, this minor project under his guidance.

We also want to express my gratitude to all the faculties at Nirma University, whose kindness and patience helped us foster a pleasant working atmosphere. We do, in fact, acknowledge the assistance of these people with a great deal of joy and thanks.

My friends and the department colleagues who assisted in the successful completion of this minor project have our sincere gratitude.

ABSTRACT/ Outline

The application of computer vision techniques to the image retrieval problem—that is, the challenge of finding digital pictures in vast databases—is known as content-based image retrieval, query by image content, and content-based visual information retrieval. "Content-based" means that the search analyzes the contents of the image and the metadata such as keywords, tags, or descriptions associated with the particular image. The term "content" in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself. Content Based Image Retrieval is desirable because searches that rely purely on metadata are dependent on annotation quality and completeness. So similar images are retrieved based on their content similarity. So we have chosen the flutter framework for app development because flutter lets you export the same code for Android as well as IOS.

CONTENTS

Chapter 1 Introduction

1.1 General

Search by image content, content-based visual information retrieval, and content-based image retrieval are all terms used to describe the application of computer vision techniques to the image retrieval issue or the difficulty of locating digital photos in enormous databases. "Content-based" refers to a search that examines the contents of the picture as well as any related metadata, such as tags, keywords, or descriptions. Colors, forms, textures, and any other information that can be inferred from the image itself are all examples of "content" in this context. It is preferable to employ content-based image retrieval since searches that just use metadata are reliant on the accuracy and completeness of the annotations. Accordingly, similar photographs are found depending on how similar their contents are.

1.2 Objectives

- The primary objective is to search those images that are containing only textual form of data.
- Images should be searched based on the query we enter in the text field provided in the application.
- Images that do not contain any kind of textual information can be searched based on the object that are present in the image such as a cat, dog, any logo, etc.
- Application should be password protected so any authentication mechanism needs to be implemented.
- Application should let the user move or copy the resultant images in any available directory on the phone storage.
- Resultant images can be shared directly with others via any other social media app such as Whatsapp etc.

1.3 Problem Statement

The main goal is to find photos that solely include textual data in their content. Based on the search term we provide in the application's given text box, images should be found. The item that is present in a picture, such as a cat, dog, any logo, etc., may be used to search for images that do not contain any textual information. Any authentication system must be installed since applications should be password-protected. The user should be able to copy or transfer the generated photographs into any directory on the phone's storage, according to the application. Through any other social networking programme, such Whatsapp or another, the resulting photographs may be shared directly with others.

1.4 **Comparison of Various Image Retrieval Algorithms**

Reference Paper	Algorithm	Working	Advantage
A review on content-based image retrieval system: present trends and future challenges	Feature Extraction based on color and shape and Distance based on Minkowski and Euclidean	Extracts feature such as shape, color and texture and compares with features of other image using different distance measures.	Accurate searching of images in database as it considers various parameters.
CONTENT BASED IMAGE RETRIEVAL	Based on SURF and BRIEF Algorithm	SURF: Creates a grid around the key point. BRIEF: Feature detection, description and matching	SURF: Uses a box filter approximation. BRIEF: High recognition rate
Content-Based Image Retrieval - Approaches and Trends of the New Age	Annotation and concept detection	Allows image search and concept detection through supervised classification that uses concepts such as city, landscape and forest.	Higher accuracy for text based and scenario-based searching.
Performance evaluation in content-based image retrieval overview and proposals.	Textual information retrieval (TREC)	Comparison based on relevance of image.	Accurate for text-based retrieval

Chapter 2 Literature survey on technologies

2.1 Why Flutter?

On desktop, mobile, and web platforms, Flutter programming can run. Consequently, you do not need to employ developers for every platform. With Flutter, you simply need to write the code once, and you can be sure that the app will function on all other platforms. So Flutter is affordable.

Because Flutter just requires a single code update, adding features to your app is quick.

- Other functionalities that flutter harnesses are:
- Flutter is Cross-Platform
- Has a powerful UI-Engine
- Flutter Renders UI

2.1.1 The advantages of Flutter over other Frameworks:

- Package is smaller than other available frameworks
- Uses Dart language.
- Flutter has been widely accepted and has a larger community.
- Has support for native UI elements.
- Best suited for complex applications.

2.2 About Unsplash API

Images on the internet may be used for free by visiting Unsplash. Unsplash is a platform made possible by a wonderful community that has donated hundreds of thousands of its members' original photographs to inspire creativity all across the world. The Unsplash API is a cutting-edge JSON API that presents all the data you'll need to create any kind of user experience. The idea was basic. We struggled to locate high-quality, useful pictures, which is how Unsplash was created. Unsplash is now a platform supported by a community that freely shares its work in favor of all creatives. None of this would be possible without them.

2.3 About Android Native

The Native Development Kit (NDK) is a set of tools that enables the usage of C and C++ code with Android and offers platform libraries for managing native activities and accessing physical device components like sensors and touch input. The majority of inexperienced Android programmers that must solely use Java code and framework APIs to create their apps might not find the NDK adequate.

The NDK, however, can be helpful in situations when you must perform any of the following: To achieve low latency or operate computationally demanding applications, such games or physics simulations, squeeze more performance out of a device. Reuse C or C++ libraries created by you or other developers.

2.4 About Google-ML

ML Kit is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package.With a focus on "on-device" ML APIs, ML Kit is a free machine learning solution for iOS and Android apps. It also offers a simple interface to other Google cloud AI products, including the TensorFlow Lite, Neural Network APIs, and Google Cloud Vision API.

Chapter 3 Methodology

We have implemented two kinds of methods for searching images in our app. The first method is an online method where we are performing image search and retrieving images from an online server and the second method is where we are performing image search on all the images present on the local storage of the device.

3.1 Methodology for Online Search

To perform an image search on the online platform we are using the Unsplash API provided by Unsplash. We have to just provide the search criteria for which images will be searched and send it as a request to the Unsplash servers by calling the Unsplash API. Unsplash has all its images labeled and tagged as well and compares them with the search criteria provided by the user and will return all the matched resultant images as a response from the server.

```
Future<void> onlineImageSearch(String searchString) async {
  onlineImagesUrls.clear();
  localImagesFiles.clear();

  String url =
      "https://api.unsplash.com/search/photos?page=1&query=$searchString"
      "&client_id=GlUpzb7r1-eZ4vbhFDszmeI81YSuJPMR9kkah2QTbZQ";

  final response = await http.get(Uri.parse(url));

  Map responseData = json.decode(response.body) as Map;
  for (var imageData in responseData['results']) {
    onlineImagesUrls.add(imageData['urls']['regular']);
  }

  notifyListeners();
}
```

3.2 Methodology for Local Search

For performing image search on all the local data, we are first reading all the available images on the local data. as well as declaring a map which will contain all the text extracted from the images after performing OCR scanning on them where the Keys of the map will be the absolute paths of the images and the values will the texts recognized from the images after performing OCR as well as declaring a temporary list which will contain all the resultant matching images after performing the search query.

```
List<Map<String, dynamic>> localImagesData = [];

List<File> localImagesFiles = [];
List<String> onlineImagesUrls = [];

class ImagesProvider with ChangeNotifier {
  deleteLocalImageFile(File file) {
    if (localImagesFiles.contains(file)) {
      localImagesFiles.remove(file);
      notifyListeners();
    }
  }

  List getImages() {
    if (localImagesFiles.isEmpty) {
      return onlineImagesUrls;
    } else {
      return localImagesFiles;
    }
  }
```

Then we are iteratively comparing the values stored in the map for each image and if the values which are the texts extracted from the image matches with the search criteria then we will simply add that image to the temporary array which will contain all the resultant images.

```
  Future<void> localImageSearch(String searchString) async {
    onlineImagesUrls.clear();
    localImagesFiles.clear();
    RegExp reg = RegExp(searchString, caseSensitive: false);
    localImagesData.forEach((image) {
      if (reg.hasMatch(image['text'] as String)) {
        File imageFile = File(image['file'] as String);
        localImagesFiles.add(imageFile);
      }
    });

    notifyListeners();
  }
}
```

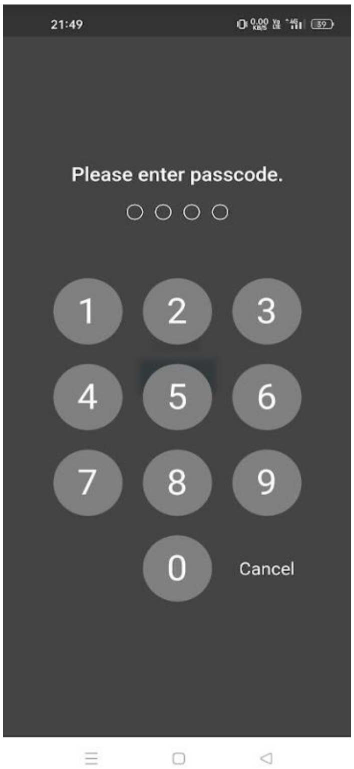
We are using google's ML Kit inbuilt method which will perform OCR on the image and will generate all the textual data found in the images. Then we are creating a list of all the textual data and then add them to the map as the values for the image which will be the key in the map.

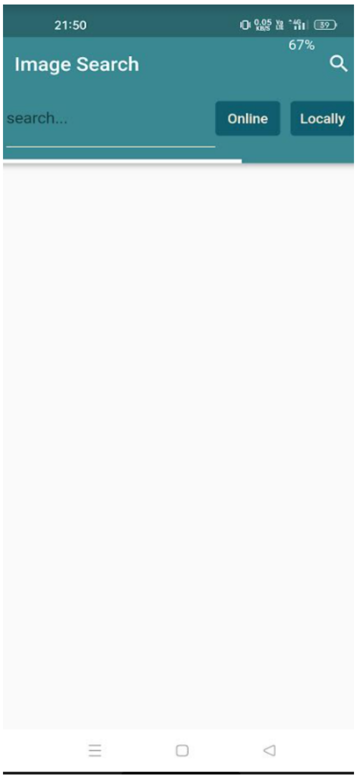
```
  int count = 0;
  totalFiles = images.length;
  print(totalFiles);
  await Future.forEach(images, (photo) async {
    File? photoFile = await photo.file;
    final InputImage inputImage = InputImage.fromFilePath(photoFile!.path);

    var recognizedText = await textRecognizer.processImage(inputImage);
    String imageText = recognizedText.text;

    final List<ImageLabel> labels =
        await imageLabeler.processImage(inputImage);

    for (ImageLabel label in labels) {
      final String text = label.label;
      imageText = imageText + " $text";
    }

    localImagesData.add({'file': photoFile.path, 'text': imageText});

    if (count > 10) {
      fileImage.writeAsStringSync(json.encode(localImagesData));
      count = 0;
    }
    count++;
    scannedFiles++;
    print(scannedFiles);
    notifyListeners();
  });

  textRecognizer.close();
  imageLabeler.close();
}
```

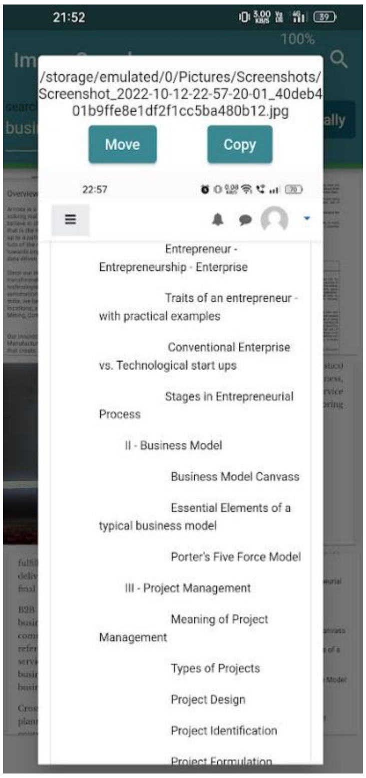
As the scanning and searching part is very time consuming we have created a JSON file that will contain all the data extracted from the images after scanning and performing OCR on them. So that the next time when the app is launched all the images will not be scanned again and only those which are new and not present in the JSON file will be scanned which will significantly reduce the processing time and searching will be highly omptimized.

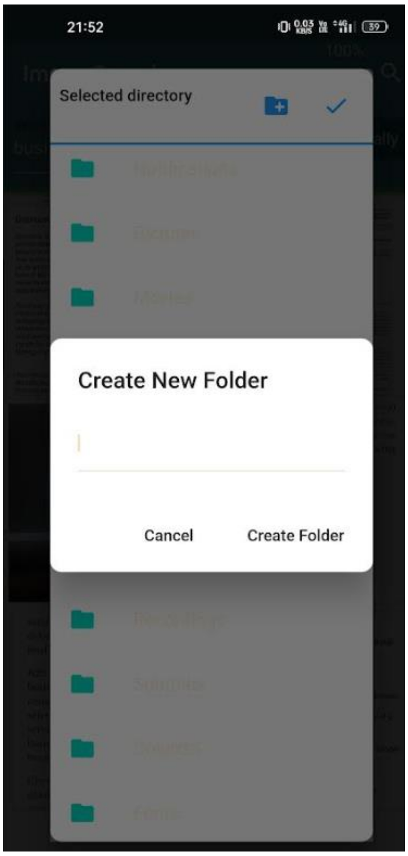
```
class LocalImageScanning with ChangeNotifier {
  localImageScanning() async {

    var textRecognizer = ml.GoogleMlKit.vision.textRecognizer();
    final ImageLabelerOptions options =
        ImageLabelerOptions(confidenceThreshold: 0.6);
    final imageLabeler = ImageLabeler(options: options);
    Directory dir = await syspath.getApplicationDocumentsDirectory();
    File fileImage = File("${dir.path}/savedScannedFiles.json");
    List<AssetEntity> images = [];  // images to be scanned right now
    List<String> storedFilesPaths = []; //paths for files already scanned in device

    if (fileImage.existsSync()) {  //adding already scanned files to the app data
      final data = fileImage.readAsStringSync();
      final photos = json.decode(data) as List;
      photos.forEach((photo) {
        localImagesData.add(photo);
      });
    }

    //extracting path data from already scanned files
    localImagesData.forEach((photo) {
      storedFilesPaths.add(path.basename(photo['file'] as String));
    });

    final List<AssetPathEntity> paths = await PhotoManager.getAssetPathList(
        type: RequestType.image, hasAll: false);

    //comparing paths and only adding those files to images variable which were not scanned
    await Future.forEach(paths, (folder) async {
      var photos = await folder.getAssetListRange(start: 0, end: 100);

      await Future.forEach(photos, (photo) {
        if (!storedFilesPaths.contains(photo.title)) {
          images.add(photo);
        }
      });
    });
```

Chapter 4 Final Application

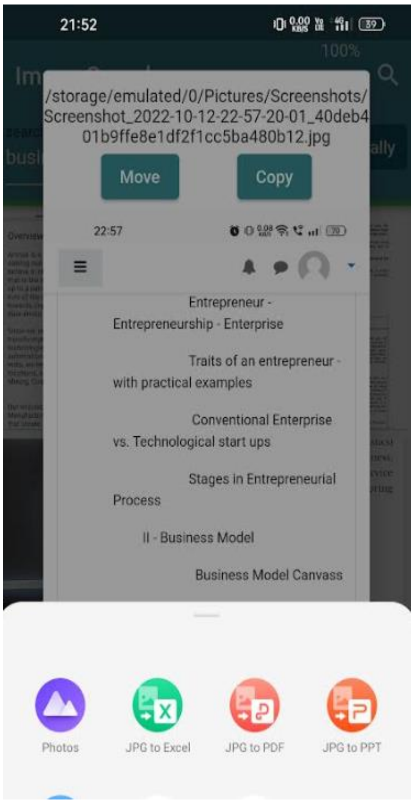







21:52
Selected directory
Create New Folder
Cancel
Create Folder


21:53
100%
Image Search
carrot
Online
Locally


21:52
/storage/emulated/0/Pictures/Screenshots/Screenshot_2022-10-12-22-57-20-01_40deb401b9ffe8e1df2f1cc5ba480b12.jpg
Move
Copy
22:57
Entrepreneur - Entrepreneurship - Enterprise
Traits of an entrepreneur - with practical examples
Conventional Enterprise vs. Technological start ups
Stages in Entrepreneurial Process
II - Business Model
Business Model Canvass
Photos
JPG to Excel
JPG to PDF
JPG to PPT

Chapter-F Summary, Conclusion and Future Scope

Summary

The drawbacks of metadata-based systems and the wide range of applications for effective image retrieval have increased interest in CBIR. Existing technology makes it simple to search for textual descriptions of photographs, but doing so requires humans to manually annotate each image in the database. For really big datasets or automatically generated photographs, such as those from security cameras, this may not be practicable. Additionally, it is easy to overlook pictures whose descriptions make use of several synonyms. Systems that classify photos according to semantic classes, such as "cat" as a subclass of "animal," can avoid the miscategorization issue, but also make it harder for users to locate images that might actually be "cats" but are only categorized as "animals." There are many standards for categorizing photos, but scaling and miscategorization problems persist in all of them.

The first CBIR systems were created to search databases using the color, texture, and shape characteristics of images. User-friendly interfaces were clearly needed after these technologies were created. In order to better serve the needs of the user conducting the search, human-centered design initiatives in the CBIR field began to be implemented.

Conclusion

Image retrieval is the task of retrieving relevant images in image databases which has had an important role in different medical, economic and other fields recently. The first ideas were the retrieving of images based on text describers which were created by humans, without considering their visual characteristics. CBIR is a search method using the structural and conceptual characteristics of the image. Due to the increasing demand for the optimum retrieval of images in large image databases, research in the field of image retrieval has gained high attention. In general, an image retrieval system is divided into three parts of a user interface, a database, and the processing section. One of the most important achievements of image retrieval is the CBIR system.

Future Scope

The OCR algorithm can be further optimized and improvised for more efficient and effective searching so that it can give better results in less time. The Object detection algorithm can also be improved so that more objects can be retrieved from the images and can be properly classified for better search results. We can also enable the multi-threading for the app so that the background process that performs all the searching and provides the output can execute more efficiently and generates the results faster.